whale

May 19, 2019

In [8]:
```
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
import matplotlib.image as mplimg
from matplotlib.pyplot import imshow
from keras.utils import plot_model
from PIL import Image as pil_image

from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder

from keras import layers
from keras.preprocessing import image
from keras.applications.imagenet_utils import preprocess_input
from keras.layers import Input, Dense, Activation, BatchNormalization, Flatten, Conv2D
from keras.layers import AveragePooling2D, MaxPooling2D, Dropout
from keras.models import Model

import keras.backend as K
from keras.models import Sequential

import warnings
warnings.simplefilter("ignore", category=DeprecationWarning)
```

In [2]:
```
train_df = pd.read_csv("~/Downloads/humpback-whale-identification/train.csv")
train_df.head()
```

Out[2]:

	Image	Id
0	0000e88ab.jpg	w_f48451c
1	0001f9222.jpg	w_c3d896a
2	00029d126.jpg	w_20df2c5
3	00050a15a.jpg	new_whale
4	0005c1ef8.jpg	new_whale

In [6]:
```
print("Preparing images")
X_train = np.zeros((train_df.shape[0], 128, 128, 3))
count = 0
```

```
        for fig in train_df['Image']:
        #load images into images of size 128x128x3
            img = image.load_img("train/"+fig, target_size=(128, 128, 3))
            x = image.img_to_array(img)
            x = preprocess_input(x)

            X_train[count] = x
            if (count%500 == 0):
                print("Processing image: ", count+1, ", ", fig)
            count += 1
        X_train = X_train / 255.0
        print(X_train.shape)
```

```
Preparing images
Processing image:  1 ,  0000e88ab.jpg
Processing image:  501 ,  04c72257b.jpg
Processing image:  1001 ,  09cacb84d.jpg
Processing image:  1501 ,  0ef961892.jpg
Processing image:  2001 ,  141b56a1a.jpg
Processing image:  2501 ,  199a417aa.jpg
Processing image:  3001 ,  1ec170983.jpg
Processing image:  3501 ,  23f084b93.jpg
Processing image:  4001 ,  29163ad0b.jpg
Processing image:  4501 ,  2e0fab120.jpg
Processing image:  5001 ,  3347515d9.jpg
Processing image:  5501 ,  3842d71dc.jpg
Processing image:  6001 ,  3d7f4c7d5.jpg
Processing image:  6501 ,  425f763ca.jpg
Processing image:  7001 ,  4714400cd.jpg
Processing image:  7501 ,  4c082fbdf.jpg
Processing image:  8001 ,  50c683e23.jpg
Processing image:  8501 ,  560d986ad.jpg
Processing image:  9001 ,  5b68c83ed.jpg
Processing image:  9501 ,  60410f111.jpg
Processing image:  10001 ,  654951f81.jpg
Processing image:  10501 ,  6a572256c.jpg
Processing image:  11001 ,  6f96f55b6.jpg
Processing image:  11501 ,  74da2b511.jpg
Processing image:  12001 ,  7989d9a27.jpg
Processing image:  12501 ,  7e5aa2d8a.jpg
Processing image:  13001 ,  832382cfb.jpg
Processing image:  13501 ,  87f6c0a15.jpg
Processing image:  14001 ,  8cfc22e5d.jpg
Processing image:  14501 ,  91dcfedcd.jpg
Processing image:  15001 ,  97079398e.jpg
Processing image:  15501 ,  9c2ad64a9.jpg
Processing image:  16001 ,  a11956dff.jpg
```

```
Processing image:  16501 ,  a5f9ffe86.jpg
Processing image:  17001 ,  aaf1a967b.jpg
Processing image:  17501 ,  af9a1ffc6.jpg
Processing image:  18001 ,  b4e02531d.jpg
Processing image:  18501 ,  ba2355ca6.jpg
Processing image:  19001 ,  bf60e7fed.jpg
Processing image:  19501 ,  c49f39ce3.jpg
Processing image:  20001 ,  c960111d0.jpg
Processing image:  20501 ,  ce7984d8a.jpg
Processing image:  21001 ,  d38efaec9.jpg
Processing image:  21501 ,  d831d28ee.jpg
Processing image:  22001 ,  dd3ca2387.jpg
Processing image:  22501 ,  e288d66cf.jpg
Processing image:  23001 ,  e7cc793db.jpg
Processing image:  23501 ,  ec8c7229d.jpg
Processing image:  24001 ,  f1b850552.jpg
Processing image:  24501 ,  f6af8a4b8.jpg
Processing image:  25001 ,  fc09f2302.jpg
(25361, 128, 128, 3)
```

```
In [7]: for i in range(10):
            plt.imshow(X_train[i][:,:,0], cmap="gray")
            plt.title(plt.title(train_df.iloc[i,0]))
            plt.axis("off")
            plt.show()
```

Text(0.5, 1.0, '0001f9222.jpg')

Text(0.5, 1.0, '00029d126.jpg')

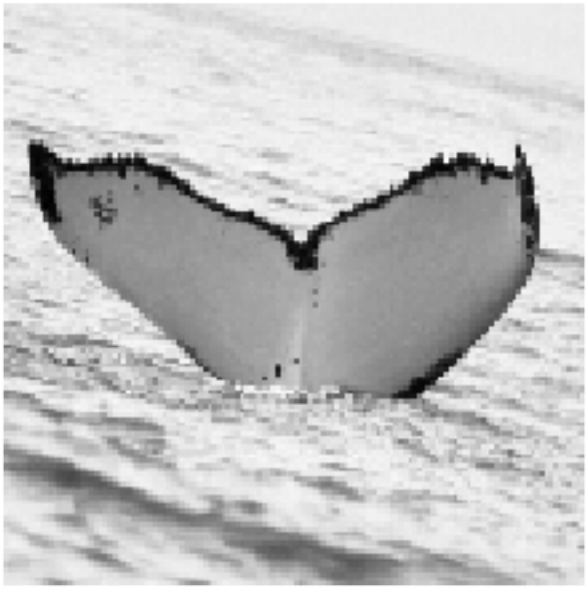
Text(0.5, 1.0, '00050a15a.jpg')

Text(0.5, 1.0, '0005c1ef8.jpg')

Text(0.5, 1.0, '0006e997e.jpg')

Text(0.5, 1.0, '000a6daec.jpg')

Text(0.5, 1.0, '000f0f2bf.jpg')



Text(0.5, 1.0, '0016b897a.jpg')


```
In [8]: values = np.array(train_df['Id'])
        label_encoder = LabelEncoder()
        integer_encoded = label_encoder.fit_transform(values)
        # print(integer_encoded)
        onehot_encoder = OneHotEncoder(sparse=False)
        integer_encoded = integer_encoded.reshape(len(integer_encoded), 1)
        onehot_encoded = onehot_encoder.fit_transform(integer_encoded)
        # print(onehot_encoded)
        y = onehot_encoded
        print(y.shape)

(25361, 5005)
```

0.1 Distribution of images per whale is highly skewed.

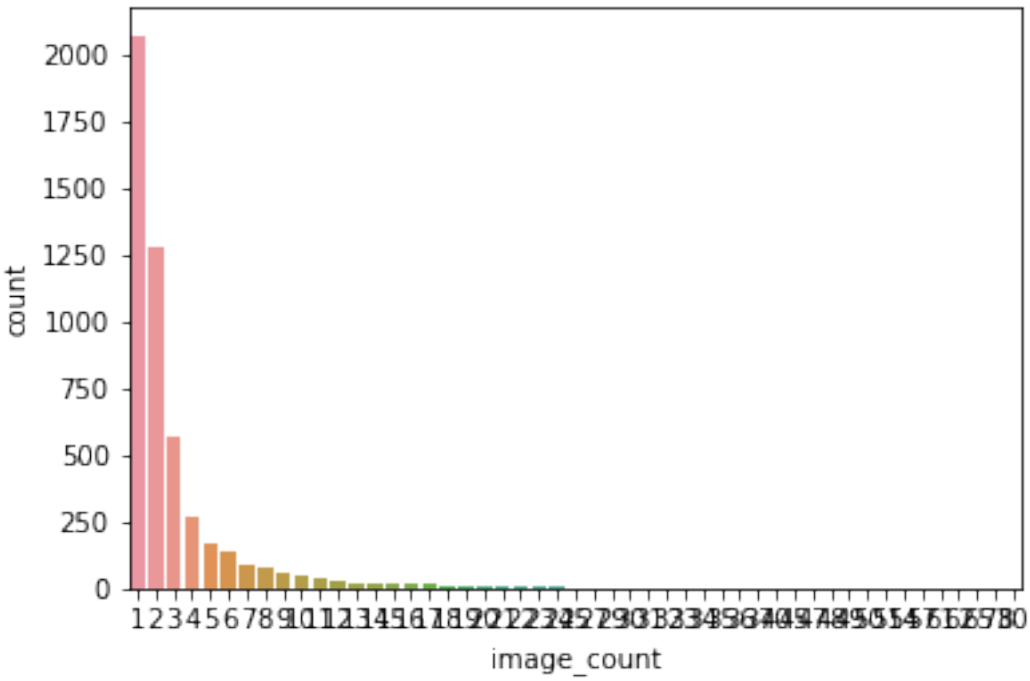
2000+ whales have just one image
Single whale with most images have 73 of them
Images dsitribution:

- almost 30% comes from whales with 4 or less images
- almost 40% comes from 'new_whale' group
- the rest 30% comes from whales with 5-73 images

```
In [21]: train_df['Id'].value_counts()[:4]
```

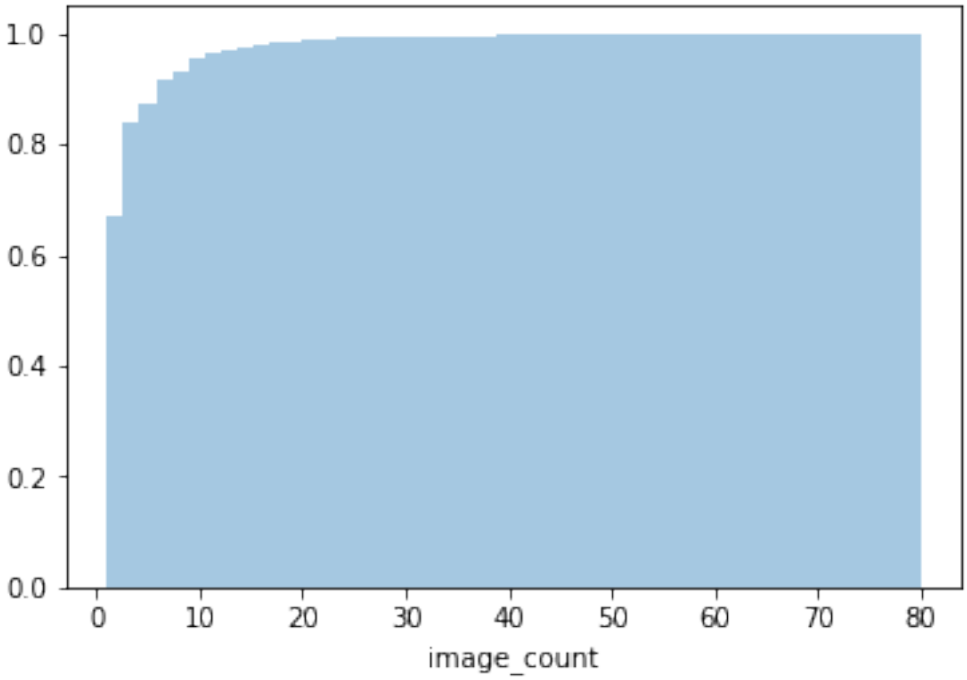
```
Out[21]: new_whale    9664
         w_23a388d      73
         w_9b5109b      65
         w_9c506f6      62
         Name: Id, dtype: int64
```

```
In [23]: import seaborn as sns
         counted = train_df.groupby("Id").count().rename(columns={"Image":"image_count"})
         counted.loc[counted["image_count"] > 80,'image_count'] = 80
         plt.figure()
         sns.countplot(data=counted, x="image_count")
         plt.show()
         sns.distplot(counted["image_count"], norm_hist=True, kde=False, hist_kws={'cumulative
```




```
/anaconda3/lib/python3.6/site-packages/matplotlib/axes/_axes.py:6521: MatplotlibDeprecationWarn
The 'normed' kwarg was deprecated in Matplotlib 2.1 and will be removed in 3.1. Use 'density' i
  alternative="'density'", removal="3.1")
```

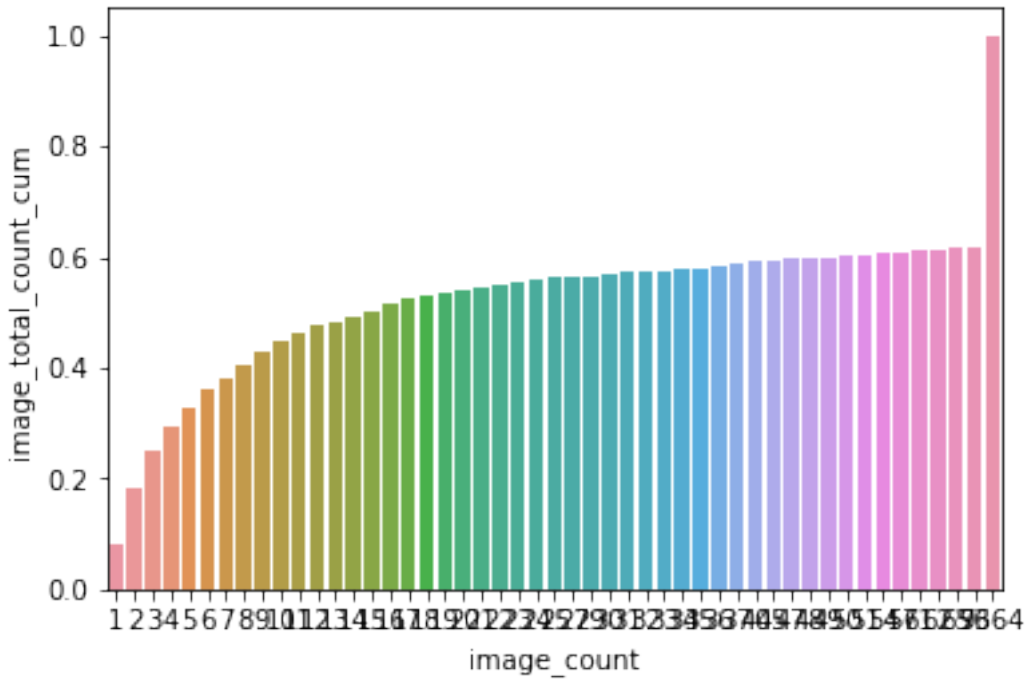
```
Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1c118c18>
```



This shows that 2000 of the 5K classes have only one image. About 1250 classes have 2 images, and so on.

```
In [26]: image_count_for_whale = train_df.groupby("Id", as_index=False).count().rename(columns=
         whale_count_for_image_count = image_count_for_whale.groupby("image_count", as_index=Fa
         whale_count_for_image_count['image_total_count'] = whale_count_for_image_count['image_
         whale_count_for_image_count['image_total_count_cum'] = whale_count_for_image_count["im
         sns.barplot(x='image_count',y='image_total_count_cum',data=whale_count_for_image_count
```

```
Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x1de607a0f0>
```




```
In [27]: whale_count_for_image_count[:5]
```

Out[27]:

	image_count	whale_count	image_total_count	image_total_count_cum
0	1	2073	2073	0.081740
1	2	1285	2570	0.183076
2	3	568	1704	0.250266
3	4	273	1092	0.293324
4	5	172	860	0.327235

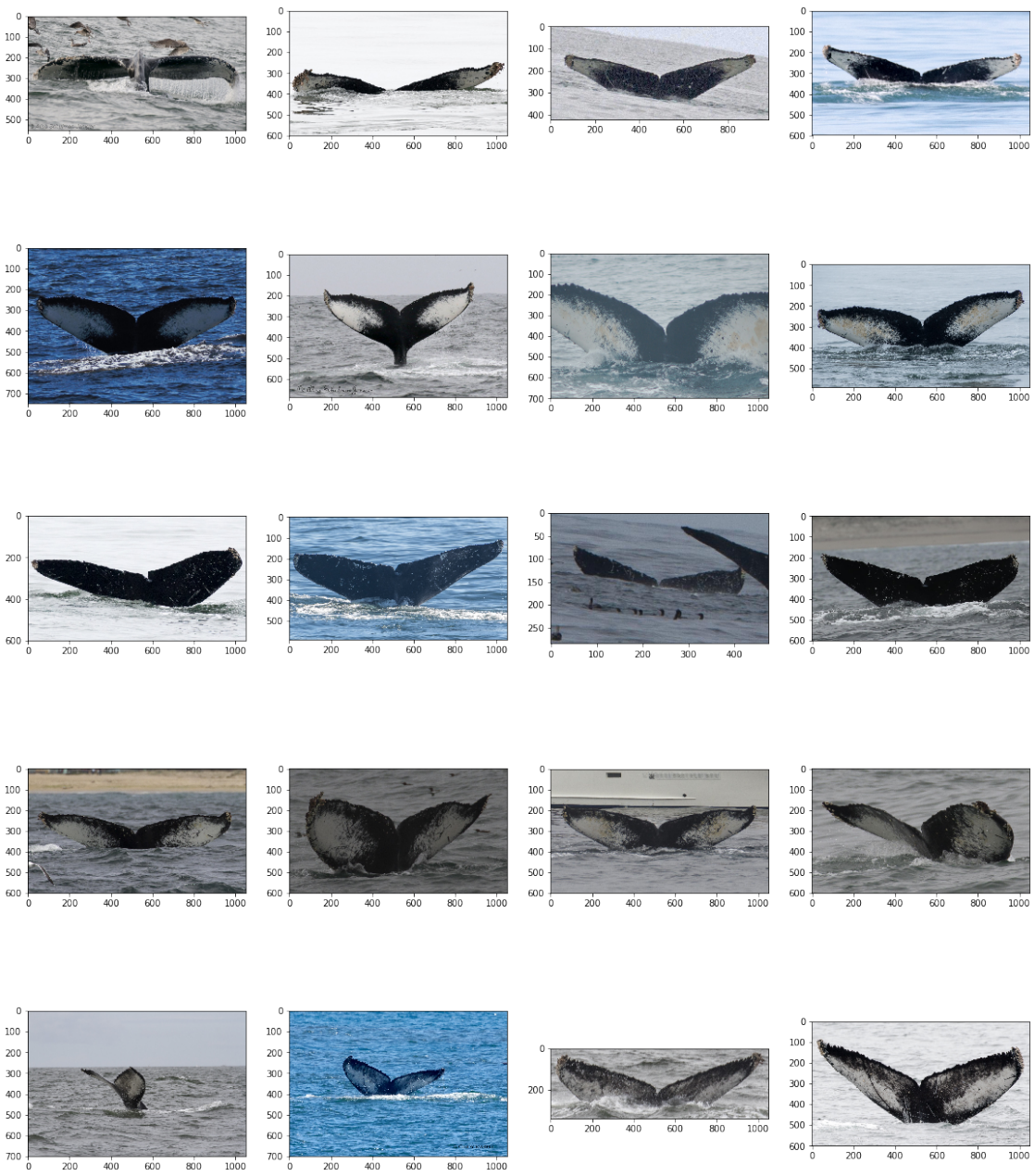
```
In [28]: whale_count_for_image_count[-3:]
```

Out[28]:

	image_count	whale_count	image_total_count	image_total_count_cum
46	65	1	65	0.616064
47	73	1	73	0.618942
48	9664	1	9664	1.000000

```
In [33]: import cv2
         topN=5
         top_whales = train_df['Id'].value_counts().index[1:1+topN]
         fig = plt.figure(figsize = (20, 5*topN))

         for widx, whale in enumerate(top_whales):
             for idx, img_name in enumerate(train_df[train_df['Id'] == whale]['Image'][:4]):
                 axes = widx*4 + idx+1
                 y = fig.add_subplot(topN, 4, axes)
```

```
        img = cv2.imread(os.path.join("","train",img_name))
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        y.imshow(img)

    plt.show()
```

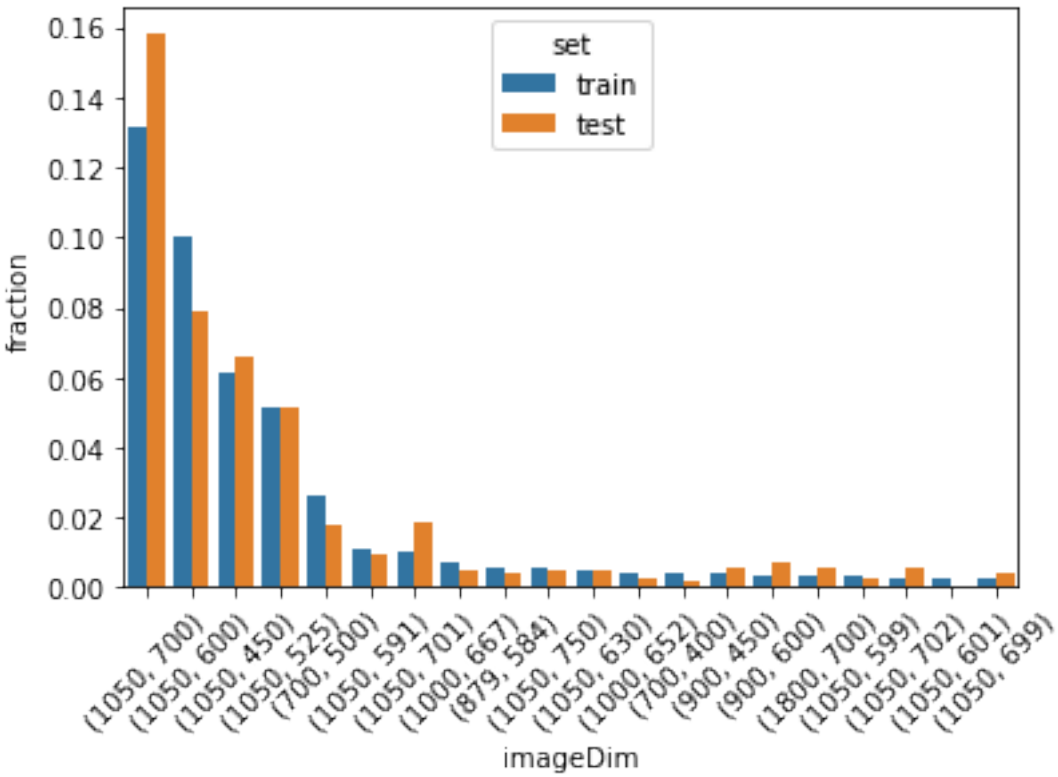


over 7000 unique resolutions but 39 most popular cover ~45% images (both in train and in test)

```
In [5]: import collections
        import os
```

```
        from PIL import Image
        imageSizes_train = collections.Counter([Image.open(f'train/{filename}').size
                                for filename in os.listdir(f"train")])
        imageSizes_test = collections.Counter([Image.open(f'test/{filename}').size
                                for filename in os.listdir(f"test")])
```

```
In [11]: import seaborn as sns
         def isdf(imageSizes):
             imageSizeFrame = pd.DataFrame(list(imageSizes.most_common()),columns = ["imageDim"
             imageSizeFrame['fraction'] = imageSizeFrame['count'] / sum(imageSizes.values())
             imageSizeFrame['count_cum'] = imageSizeFrame['count'].cumsum()
             imageSizeFrame['count_cum_fraction'] = imageSizeFrame['count_cum'] / sum(imageSize
             return imageSizeFrame

         train_isdf = isdf(imageSizes_train)
         train_isdf['set'] = 'train'
         test_isdf = isdf(imageSizes_test)
         test_isdf['set'] = 'test'

         isizes = train_isdf.merge(test_isdf, how="outer", on="imageDim")
         isizes['total_count'] = isizes['count_x'] + isizes['count_y']
         dims_order = isizes.sort_values('total_count', ascending=False)[['imageDim']]
         len(dims_order)
         isizes = pd.concat([train_isdf, test_isdf])
         popularSizes = isizes[isizes['fraction'] > 0.002]
         popularSizes.groupby('set').max()['count_cum_fraction']
         sns.barplot(x='imageDim',y='fraction',data = popularSizes, hue="set")
         _ = plt.xticks(rotation=45)
```



1 Siamese Neural Network

```
In [26]: from keras import regularizers
         from keras.optimizers import Adam
         from keras.engine.topology import Input
         from keras.layers import Activation, Add, BatchNormalization, Concatenate, Conv2D, Den
         from keras.models import Model

         img_shape    = (384,384,1) # The image shape used by the model

         def subblock(x, filter, **kwargs):
             x = BatchNormalization()(x)
             y = x
             y = Conv2D(filter, (1, 1), activation='relu', **kwargs)(y) # Reduce the number of
             y = BatchNormalization()(y)
             y = Conv2D(filter, (3, 3), activation='relu', **kwargs)(y) # Extend the feature f
             y = BatchNormalization()(y)
             y = Conv2D(K.int_shape(x)[-1], (1, 1), **kwargs)(y) # no activation # Restore the
             y = Add()([x,y]) # Add the bypass connection
             y = Activation('relu')(y)
             return y
```

```
def build_model(lr, l2, activation='sigmoid'):

    ##############
    # BRANCH MODEL
    ##############
    regul  = regularizers.l2(l2)
    optim  = Adam(lr=lr)
    kwargs = {'padding':'same', 'kernel_regularizer':regul}

    inp = Input(shape=img_shape) # 384x384x1
    x   = Conv2D(64, (9,9), strides=2, activation='relu', **kwargs)(inp)

    x   = MaxPooling2D((2, 2), strides=(2, 2))(x) # 96x96x64
    for _ in range(2):
        x = BatchNormalization()(x)
        x = Conv2D(64, (3,3), activation='relu', **kwargs)(x)

    x = MaxPooling2D((2, 2), strides=(2, 2))(x) # 48x48x64
    x = BatchNormalization()(x)
    x = Conv2D(128, (1,1), activation='relu', **kwargs)(x) # 48x48x128
    for _ in range(4): x = subblock(x, 64, **kwargs)

    x = MaxPooling2D((2, 2), strides=(2, 2))(x) # 24x24x128
    x = BatchNormalization()(x)
    x = Conv2D(256, (1,1), activation='relu', **kwargs)(x) # 24x24x256
    for _ in range(4): x = subblock(x, 64, **kwargs)

    x = MaxPooling2D((2, 2), strides=(2, 2))(x) # 12x12x256
    x = BatchNormalization()(x)
    x = Conv2D(384, (1,1), activation='relu', **kwargs)(x) # 12x12x384
    for _ in range(4): x = subblock(x, 96, **kwargs)

    x = MaxPooling2D((2, 2), strides=(2, 2))(x) # 6x6x384
    x = BatchNormalization()(x)
    x = Conv2D(512, (1,1), activation='relu', **kwargs)(x) # 6x6x512
    for _ in range(4): x = subblock(x, 128, **kwargs)

    x             = GlobalMaxPooling2D()(x) # 512
    branch_model  = Model(inp, x)

    #############
    # HEAD MODEL
    #############
    mid        = 32
    xa_inp     = Input(shape=branch_model.output_shape[1:])
    xb_inp     = Input(shape=branch_model.output_shape[1:])
    x1         = Lambda(lambda x : x[0]*x[1])([xa_inp, xb_inp])
```

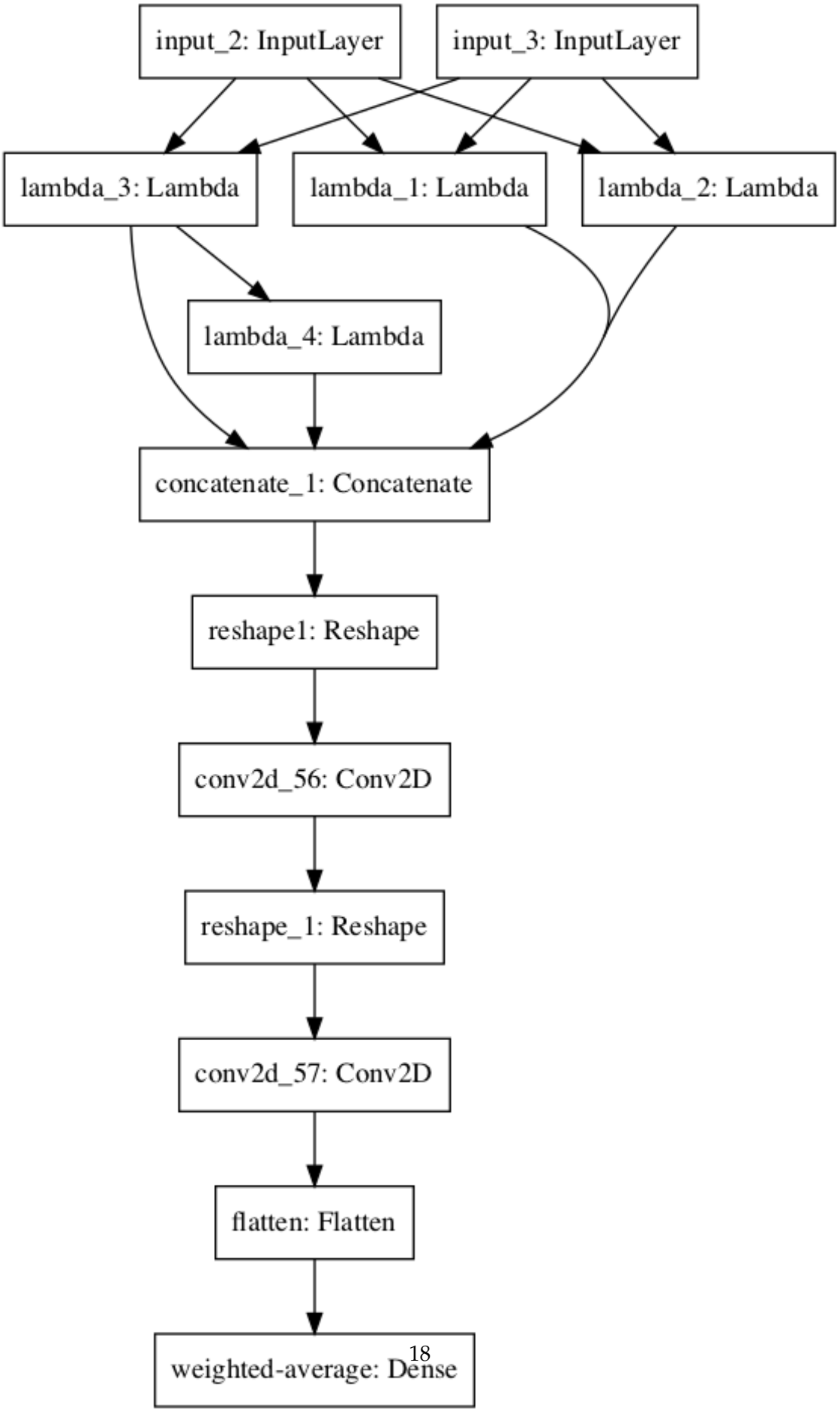
```
        x2          = Lambda(lambda x : x[0] + x[1])([xa_inp, xb_inp])
        x3          = Lambda(lambda x : K.abs(x[0] - x[1]))([xa_inp, xb_inp])
        x4          = Lambda(lambda x : K.square(x))(x3)
        x           = Concatenate()([x1, x2, x3, x4])
        x           = Reshape((4, branch_model.output_shape[1], 1), name='reshape1')(x)

        # Per feature NN with shared weight is implemented using CONV2D with appropriate
        x           = Conv2D(mid, (4, 1), activation='relu', padding='valid')(x)
        x           = Reshape((branch_model.output_shape[1], mid, 1))(x)
        x           = Conv2D(1, (1, mid), activation='linear', padding='valid')(x)
        x           = Flatten(name='flatten')(x)

        # Weighted sum implemented as a Dense layer.
        x           = Dense(1, use_bias=True, activation=activation, name='weighted-averag
        head_model = Model([xa_inp, xb_inp], x, name='head')

        ########################
        # SIAMESE NEURAL NETWORK
        ########################
        # Complete model is constructed by calling the branch model on each input image,
        # and then the head model on the resulting 512-vectors.
        img_a       = Input(shape=img_shape)
        img_b       = Input(shape=img_shape)
        xa          = branch_model(img_a)
        xb          = branch_model(img_b)
        x           = head_model([xa, xb])
        model       = Model([img_a, img_b], x)
        model.compile(optim, loss='binary_crossentropy', metrics=['binary_crossentropy',
        return model, branch_model, head_model

    model, branch_model, head_model = build_model(64e-5,0)
    head_model.summary()
```

```
__________________________________________________________________________________________
Layer (type)                 Output Shape              Param #     Connected to
==========================================================================================
input_2 (InputLayer)         (None, 512)               0
__________________________________________________________________________________________
input_3 (InputLayer)         (None, 512)               0
__________________________________________________________________________________________
lambda_3 (Lambda)            (None, 512)               0           input_2[0][0]
                                                                   input_3[0][0]
__________________________________________________________________________________________
lambda_1 (Lambda)            (None, 512)               0           input_2[0][0]
                                                                   input_3[0][0]
__________________________________________________________________________________________
lambda_2 (Lambda)            (None, 512)               0           input_2[0][0]
                                                                   input_3[0][0]
```

```
____________________________________________________________________________________________________
lambda_4 (Lambda)                (None, 512)           0           lambda_3[0][0]
____________________________________________________________________________________________________
concatenate_1 (Concatenate)      (None, 2048)          0           lambda_1[0][0]
                                                                   lambda_2[0][0]
                                                                   lambda_3[0][0]
                                                                   lambda_4[0][0]
____________________________________________________________________________________________________
reshape1 (Reshape)               (None, 4, 512, 1)     0           concatenate_1[0][0]
____________________________________________________________________________________________________
conv2d_56 (Conv2D)               (None, 1, 512, 32)    160         reshape1[0][0]
____________________________________________________________________________________________________
reshape_1 (Reshape)              (None, 512, 32, 1)    0           conv2d_56[0][0]
____________________________________________________________________________________________________
conv2d_57 (Conv2D)               (None, 512, 1, 1)     33          reshape_1[0][0]
____________________________________________________________________________________________________
flatten (Flatten)                (None, 512)           0           conv2d_57[0][0]
____________________________________________________________________________________________________
weighted-average (Dense)         (None, 1)             513         flatten[0][0]
====================================================================================================
Total params: 706
Trainable params: 706
Non-trainable params: 0
____________________________________________________________________________________________________
```

```
In [27]: plot_model(head_model, to_file='head-model.png')
         pil_image.open('head-model.png')
```

Out[27]:


input_2: InputLayer
input_3: InputLayer
lambda_3: Lambda
lambda_1: Lambda
lambda_2: Lambda
lambda_4: Lambda
concatenate_1: Concatenate
reshape1: Reshape
conv2d_56: Conv2D
reshape_1: Reshape
conv2d_57: Conv2D
flatten: Flatten
weighted-average: Dense

```
In [28]: branch_model.summary()
```

```
_________________________________________________________________________________________________
Layer (type)                 Output Shape              Param #     Connected to
=================================================================================================
input_1 (InputLayer)         (None, 384, 384, 1)       0
_________________________________________________________________________________________________
conv2d_1 (Conv2D)            (None, 192, 192, 64)      5248        input_1[0][0]
_________________________________________________________________________________________________
max_pooling2d_1 (MaxPooling2D)  (None, 96, 96, 64)     0           conv2d_1[0][0]
_________________________________________________________________________________________________
batch_normalization_1 (BatchNor (None, 96, 96, 64)     256         max_pooling2d_1[0][0]
_________________________________________________________________________________________________
conv2d_2 (Conv2D)            (None, 96, 96, 64)        36928       batch_normalization_1[0][0]
_________________________________________________________________________________________________
batch_normalization_2 (BatchNor (None, 96, 96, 64)     256         conv2d_2[0][0]
_________________________________________________________________________________________________
conv2d_3 (Conv2D)            (None, 96, 96, 64)        36928       batch_normalization_2[0][0]
_________________________________________________________________________________________________
max_pooling2d_2 (MaxPooling2D)  (None, 48, 48, 64)     0           conv2d_3[0][0]
_________________________________________________________________________________________________
batch_normalization_3 (BatchNor (None, 48, 48, 64)     256         max_pooling2d_2[0][0]
_________________________________________________________________________________________________
conv2d_4 (Conv2D)            (None, 48, 48, 128)       8320        batch_normalization_3[0][0]
_________________________________________________________________________________________________
batch_normalization_4 (BatchNor (None, 48, 48, 128)    512         conv2d_4[0][0]
_________________________________________________________________________________________________
conv2d_5 (Conv2D)            (None, 48, 48, 64)        8256        batch_normalization_4[0][0]
_________________________________________________________________________________________________
batch_normalization_5 (BatchNor (None, 48, 48, 64)     256         conv2d_5[0][0]
_________________________________________________________________________________________________
conv2d_6 (Conv2D)            (None, 48, 48, 64)        36928       batch_normalization_5[0][0]
_________________________________________________________________________________________________
batch_normalization_6 (BatchNor (None, 48, 48, 64)     256         conv2d_6[0][0]
_________________________________________________________________________________________________
conv2d_7 (Conv2D)            (None, 48, 48, 128)       8320        batch_normalization_6[0][0]
_________________________________________________________________________________________________
add_1 (Add)                  (None, 48, 48, 128)       0           batch_normalization_4[0][0]
                                                                   conv2d_7[0][0]
_________________________________________________________________________________________________
activation_1 (Activation)    (None, 48, 48, 128)       0           add_1[0][0]
_________________________________________________________________________________________________
batch_normalization_7 (BatchNor (None, 48, 48, 128)    512         activation_1[0][0]
_________________________________________________________________________________________________
conv2d_8 (Conv2D)            (None, 48, 48, 64)        8256        batch_normalization_7[0][0]
```

```
batch_normalization_8 (BatchNor (None, 48, 48, 64)   256         conv2d_8[0][0]
conv2d_9 (Conv2D)               (None, 48, 48, 64)   36928       batch_normalization_8[0][0]
batch_normalization_9 (BatchNor (None, 48, 48, 64)   256         conv2d_9[0][0]
conv2d_10 (Conv2D)              (None, 48, 48, 128)  8320        batch_normalization_9[0][0]
add_2 (Add)                     (None, 48, 48, 128)  0           batch_normalization_7[0][0]
                                                                 conv2d_10[0][0]
activation_2 (Activation)       (None, 48, 48, 128)  0           add_2[0][0]
batch_normalization_10 (BatchNo (None, 48, 48, 128)  512         activation_2[0][0]
conv2d_11 (Conv2D)              (None, 48, 48, 64)   8256        batch_normalization_10[0][0]
batch_normalization_11 (BatchNo (None, 48, 48, 64)   256         conv2d_11[0][0]
conv2d_12 (Conv2D)              (None, 48, 48, 64)   36928       batch_normalization_11[0][0]
batch_normalization_12 (BatchNo (None, 48, 48, 64)   256         conv2d_12[0][0]
conv2d_13 (Conv2D)              (None, 48, 48, 128)  8320        batch_normalization_12[0][0]
add_3 (Add)                     (None, 48, 48, 128)  0           batch_normalization_10[0][0]
                                                                 conv2d_13[0][0]
activation_3 (Activation)       (None, 48, 48, 128)  0           add_3[0][0]
batch_normalization_13 (BatchNo (None, 48, 48, 128)  512         activation_3[0][0]
conv2d_14 (Conv2D)              (None, 48, 48, 64)   8256        batch_normalization_13[0][0]
batch_normalization_14 (BatchNo (None, 48, 48, 64)   256         conv2d_14[0][0]
conv2d_15 (Conv2D)              (None, 48, 48, 64)   36928       batch_normalization_14[0][0]
batch_normalization_15 (BatchNo (None, 48, 48, 64)   256         conv2d_15[0][0]
conv2d_16 (Conv2D)              (None, 48, 48, 128)  8320        batch_normalization_15[0][0]
add_4 (Add)                     (None, 48, 48, 128)  0           batch_normalization_13[0][0]
                                                                 conv2d_16[0][0]
activation_4 (Activation)       (None, 48, 48, 128)  0           add_4[0][0]
```

```
max_pooling2d_3 (MaxPooling2D)  (None, 24, 24, 128)  0           activation_4[0][0]
__________________________________________________________________________________________
batch_normalization_16 (BatchNo (None, 24, 24, 128)  512         max_pooling2d_3[0][0]
__________________________________________________________________________________________
conv2d_17 (Conv2D)              (None, 24, 24, 256)  33024       batch_normalization_16[0][0]
__________________________________________________________________________________________
batch_normalization_17 (BatchNo (None, 24, 24, 256)  1024        conv2d_17[0][0]
__________________________________________________________________________________________
conv2d_18 (Conv2D)              (None, 24, 24, 64)   16448       batch_normalization_17[0][0]
__________________________________________________________________________________________
batch_normalization_18 (BatchNo (None, 24, 24, 64)   256         conv2d_18[0][0]
__________________________________________________________________________________________
conv2d_19 (Conv2D)              (None, 24, 24, 64)   36928       batch_normalization_18[0][0]
__________________________________________________________________________________________
batch_normalization_19 (BatchNo (None, 24, 24, 64)   256         conv2d_19[0][0]
__________________________________________________________________________________________
conv2d_20 (Conv2D)              (None, 24, 24, 256)  16640       batch_normalization_19[0][0]
__________________________________________________________________________________________
add_5 (Add)                     (None, 24, 24, 256)  0           batch_normalization_17[0][0]
                                                                 conv2d_20[0][0]
__________________________________________________________________________________________
activation_5 (Activation)       (None, 24, 24, 256)  0           add_5[0][0]
__________________________________________________________________________________________
batch_normalization_20 (BatchNo (None, 24, 24, 256)  1024        activation_5[0][0]
__________________________________________________________________________________________
conv2d_21 (Conv2D)              (None, 24, 24, 64)   16448       batch_normalization_20[0][0]
__________________________________________________________________________________________
batch_normalization_21 (BatchNo (None, 24, 24, 64)   256         conv2d_21[0][0]
__________________________________________________________________________________________
conv2d_22 (Conv2D)              (None, 24, 24, 64)   36928       batch_normalization_21[0][0]
__________________________________________________________________________________________
batch_normalization_22 (BatchNo (None, 24, 24, 64)   256         conv2d_22[0][0]
__________________________________________________________________________________________
conv2d_23 (Conv2D)              (None, 24, 24, 256)  16640       batch_normalization_22[0][0]
__________________________________________________________________________________________
add_6 (Add)                     (None, 24, 24, 256)  0           batch_normalization_20[0][0]
                                                                 conv2d_23[0][0]
__________________________________________________________________________________________
activation_6 (Activation)       (None, 24, 24, 256)  0           add_6[0][0]
__________________________________________________________________________________________
batch_normalization_23 (BatchNo (None, 24, 24, 256)  1024        activation_6[0][0]
__________________________________________________________________________________________
conv2d_24 (Conv2D)              (None, 24, 24, 64)   16448       batch_normalization_23[0][0]
__________________________________________________________________________________________
batch_normalization_24 (BatchNo (None, 24, 24, 64)   256         conv2d_24[0][0]
__________________________________________________________________________________________
conv2d_25 (Conv2D)              (None, 24, 24, 64)   36928       batch_normalization_24[0][0]
__________________________________________________________________________________________
```

```
batch_normalization_25 (BatchNo (None, 24, 24, 64)   256         conv2d_25[0][0]
__________________________________________________________________________________________
conv2d_26 (Conv2D)              (None, 24, 24, 256)  16640       batch_normalization_25[0][0]
__________________________________________________________________________________________
add_7 (Add)                     (None, 24, 24, 256)  0           batch_normalization_23[0][0]
                                                                 conv2d_26[0][0]
__________________________________________________________________________________________
activation_7 (Activation)       (None, 24, 24, 256)  0           add_7[0][0]
__________________________________________________________________________________________
batch_normalization_26 (BatchNo (None, 24, 24, 256)  1024        activation_7[0][0]
__________________________________________________________________________________________
conv2d_27 (Conv2D)              (None, 24, 24, 64)   16448       batch_normalization_26[0][0]
__________________________________________________________________________________________
batch_normalization_27 (BatchNo (None, 24, 24, 64)   256         conv2d_27[0][0]
__________________________________________________________________________________________
conv2d_28 (Conv2D)              (None, 24, 24, 64)   36928       batch_normalization_27[0][0]
__________________________________________________________________________________________
batch_normalization_28 (BatchNo (None, 24, 24, 64)   256         conv2d_28[0][0]
__________________________________________________________________________________________
conv2d_29 (Conv2D)              (None, 24, 24, 256)  16640       batch_normalization_28[0][0]
__________________________________________________________________________________________
add_8 (Add)                     (None, 24, 24, 256)  0           batch_normalization_26[0][0]
                                                                 conv2d_29[0][0]
__________________________________________________________________________________________
activation_8 (Activation)       (None, 24, 24, 256)  0           add_8[0][0]
__________________________________________________________________________________________
max_pooling2d_4 (MaxPooling2D)  (None, 12, 12, 256)  0           activation_8[0][0]
__________________________________________________________________________________________
batch_normalization_29 (BatchNo (None, 12, 12, 256)  1024        max_pooling2d_4[0][0]
__________________________________________________________________________________________
conv2d_30 (Conv2D)              (None, 12, 12, 384)  98688       batch_normalization_29[0][0]
__________________________________________________________________________________________
batch_normalization_30 (BatchNo (None, 12, 12, 384)  1536        conv2d_30[0][0]
__________________________________________________________________________________________
conv2d_31 (Conv2D)              (None, 12, 12, 96)   36960       batch_normalization_30[0][0]
__________________________________________________________________________________________
batch_normalization_31 (BatchNo (None, 12, 12, 96)   384         conv2d_31[0][0]
__________________________________________________________________________________________
conv2d_32 (Conv2D)              (None, 12, 12, 96)   83040       batch_normalization_31[0][0]
__________________________________________________________________________________________
batch_normalization_32 (BatchNo (None, 12, 12, 96)   384         conv2d_32[0][0]
__________________________________________________________________________________________
conv2d_33 (Conv2D)              (None, 12, 12, 384)  37248       batch_normalization_32[0][0]
__________________________________________________________________________________________
add_9 (Add)                     (None, 12, 12, 384)  0           batch_normalization_30[0][0]
                                                                 conv2d_33[0][0]
__________________________________________________________________________________________
activation_9 (Activation)       (None, 12, 12, 384)  0           add_9[0][0]
```

```
batch_normalization_33 (BatchNo (None, 12, 12, 384)  1536        activation_9[0][0]
conv2d_34 (Conv2D)              (None, 12, 12, 96)   36960       batch_normalization_33[0][0]
batch_normalization_34 (BatchNo (None, 12, 12, 96)   384         conv2d_34[0][0]
conv2d_35 (Conv2D)              (None, 12, 12, 96)   83040       batch_normalization_34[0][0]
batch_normalization_35 (BatchNo (None, 12, 12, 96)   384         conv2d_35[0][0]
conv2d_36 (Conv2D)              (None, 12, 12, 384)  37248       batch_normalization_35[0][0]
add_10 (Add)                    (None, 12, 12, 384)  0           batch_normalization_33[0][0]
                                                                 conv2d_36[0][0]
activation_10 (Activation)      (None, 12, 12, 384)  0           add_10[0][0]
batch_normalization_36 (BatchNo (None, 12, 12, 384)  1536        activation_10[0][0]
conv2d_37 (Conv2D)              (None, 12, 12, 96)   36960       batch_normalization_36[0][0]
batch_normalization_37 (BatchNo (None, 12, 12, 96)   384         conv2d_37[0][0]
conv2d_38 (Conv2D)              (None, 12, 12, 96)   83040       batch_normalization_37[0][0]
batch_normalization_38 (BatchNo (None, 12, 12, 96)   384         conv2d_38[0][0]
conv2d_39 (Conv2D)              (None, 12, 12, 384)  37248       batch_normalization_38[0][0]
add_11 (Add)                    (None, 12, 12, 384)  0           batch_normalization_36[0][0]
                                                                 conv2d_39[0][0]
activation_11 (Activation)      (None, 12, 12, 384)  0           add_11[0][0]
batch_normalization_39 (BatchNo (None, 12, 12, 384)  1536        activation_11[0][0]
conv2d_40 (Conv2D)              (None, 12, 12, 96)   36960       batch_normalization_39[0][0]
batch_normalization_40 (BatchNo (None, 12, 12, 96)   384         conv2d_40[0][0]
conv2d_41 (Conv2D)              (None, 12, 12, 96)   83040       batch_normalization_40[0][0]
batch_normalization_41 (BatchNo (None, 12, 12, 96)   384         conv2d_41[0][0]
conv2d_42 (Conv2D)              (None, 12, 12, 384)  37248       batch_normalization_41[0][0]
add_12 (Add)                    (None, 12, 12, 384)  0           batch_normalization_39[0][0]
```

```
                                                                 conv2d_42[0][0]
__________________________________________________________________________________________________
activation_12 (Activation)      (None, 12, 12, 384)  0           add_12[0][0]
__________________________________________________________________________________________________
max_pooling2d_5 (MaxPooling2D)  (None, 6, 6, 384)    0           activation_12[0][0]
__________________________________________________________________________________________________
batch_normalization_42 (BatchNo (None, 6, 6, 384)    1536        max_pooling2d_5[0][0]
__________________________________________________________________________________________________
conv2d_43 (Conv2D)              (None, 6, 6, 512)    197120      batch_normalization_42[0][0]
__________________________________________________________________________________________________
batch_normalization_43 (BatchNo (None, 6, 6, 512)    2048        conv2d_43[0][0]
__________________________________________________________________________________________________
conv2d_44 (Conv2D)              (None, 6, 6, 128)    65664       batch_normalization_43[0][0]
__________________________________________________________________________________________________
batch_normalization_44 (BatchNo (None, 6, 6, 128)    512         conv2d_44[0][0]
__________________________________________________________________________________________________
conv2d_45 (Conv2D)              (None, 6, 6, 128)    147584      batch_normalization_44[0][0]
__________________________________________________________________________________________________
batch_normalization_45 (BatchNo (None, 6, 6, 128)    512         conv2d_45[0][0]
__________________________________________________________________________________________________
conv2d_46 (Conv2D)              (None, 6, 6, 512)    66048       batch_normalization_45[0][0]
__________________________________________________________________________________________________
add_13 (Add)                    (None, 6, 6, 512)    0           batch_normalization_43[0][0]
                                                                 conv2d_46[0][0]
__________________________________________________________________________________________________
activation_13 (Activation)      (None, 6, 6, 512)    0           add_13[0][0]
__________________________________________________________________________________________________
batch_normalization_46 (BatchNo (None, 6, 6, 512)    2048        activation_13[0][0]
__________________________________________________________________________________________________
conv2d_47 (Conv2D)              (None, 6, 6, 128)    65664       batch_normalization_46[0][0]
__________________________________________________________________________________________________
batch_normalization_47 (BatchNo (None, 6, 6, 128)    512         conv2d_47[0][0]
__________________________________________________________________________________________________
conv2d_48 (Conv2D)              (None, 6, 6, 128)    147584      batch_normalization_47[0][0]
__________________________________________________________________________________________________
batch_normalization_48 (BatchNo (None, 6, 6, 128)    512         conv2d_48[0][0]
__________________________________________________________________________________________________
conv2d_49 (Conv2D)              (None, 6, 6, 512)    66048       batch_normalization_48[0][0]
__________________________________________________________________________________________________
add_14 (Add)                    (None, 6, 6, 512)    0           batch_normalization_46[0][0]
                                                                 conv2d_49[0][0]
__________________________________________________________________________________________________
activation_14 (Activation)      (None, 6, 6, 512)    0           add_14[0][0]
__________________________________________________________________________________________________
batch_normalization_49 (BatchNo (None, 6, 6, 512)    2048        activation_14[0][0]
__________________________________________________________________________________________________
conv2d_50 (Conv2D)              (None, 6, 6, 128)    65664       batch_normalization_49[0][0]
__________________________________________________________________________________________________
```

```
batch_normalization_50 (BatchNo (None, 6, 6, 128)    512         conv2d_50[0][0]
__________________________________________________________________________________________________
conv2d_51 (Conv2D)              (None, 6, 6, 128)    147584      batch_normalization_50[0][0]
__________________________________________________________________________________________________
batch_normalization_51 (BatchNo (None, 6, 6, 128)    512         conv2d_51[0][0]
__________________________________________________________________________________________________
conv2d_52 (Conv2D)              (None, 6, 6, 512)    66048       batch_normalization_51[0][0]
__________________________________________________________________________________________________
add_15 (Add)                    (None, 6, 6, 512)    0           batch_normalization_49[0][0]
                                                                 conv2d_52[0][0]
__________________________________________________________________________________________________
activation_15 (Activation)      (None, 6, 6, 512)    0           add_15[0][0]
__________________________________________________________________________________________________
batch_normalization_52 (BatchNo (None, 6, 6, 512)    2048        activation_15[0][0]
__________________________________________________________________________________________________
conv2d_53 (Conv2D)              (None, 6, 6, 128)    65664       batch_normalization_52[0][0]
__________________________________________________________________________________________________
batch_normalization_53 (BatchNo (None, 6, 6, 128)    512         conv2d_53[0][0]
__________________________________________________________________________________________________
conv2d_54 (Conv2D)              (None, 6, 6, 128)    147584      batch_normalization_53[0][0]
__________________________________________________________________________________________________
batch_normalization_54 (BatchNo (None, 6, 6, 128)    512         conv2d_54[0][0]
__________________________________________________________________________________________________
conv2d_55 (Conv2D)              (None, 6, 6, 512)    66048       batch_normalization_54[0][0]
__________________________________________________________________________________________________
add_16 (Add)                    (None, 6, 6, 512)    0           batch_normalization_52[0][0]
                                                                 conv2d_55[0][0]
__________________________________________________________________________________________________
activation_16 (Activation)      (None, 6, 6, 512)    0           add_16[0][0]
__________________________________________________________________________________________________
global_max_pooling2d_1 (GlobalM (None, 512)          0           activation_16[0][0]
==================================================================================================
Total params: 2,692,096
Trainable params: 2,674,304
Non-trainable params: 17,792
__________________________________________________________________________________________________
```

```
In [29]: plot_model(branch_model, to_file='branch-model.png')
         img = pil_image.open('branch-model.png')
         img.resize([x//2 for x in img.size])
```

Out[29]: